```
Discussion : 2020-07-20
1
 2
 3
    Saurab
 4
 5
       1.
 6
         2. gram indices: abcus abc -> 450, bcu -> 679
 7
         vector dimension: 1 5000
8
9
         grams [abc, bcu, cus, ab, bc, cu, us]
10
         to avoid iterating from 1 to 5000.
11
12
    Shankar
13
14
       1. Index Map:
         2. Completed without query search in relevant words.
15
16
         3. How to implement with relevant words ?
17
18
         vector of entry \Rightarrow n = 15
19
         no. of words =>
                            1000 = n + k
20
21
        (n + k) * n for each wrong word.
22
         n^2 + nk => 0(n^2)
23
24
         Inverted index =>
25
             g1 -> [w1, w203, w500]
26
             g2 \rightarrow [w1, w609, w999]
27
28
         During Search:
29
30
         wrong\_word wk = [g100, g2, g890, g350]
31
32
         wk -> map with inverted_index -> List of List
33
         filter distinct -> list of words (relevant)
34
35
36
   What is log function ?
    2 ^ 5 = 32
37
     log2(32) = 5
38
39
40
    -----
41
42
    Minu
43
    -----
44

    flow completed.

45
        2.
46
47
    w1: apple => [ap, pp, pl, le, app, ppl, ple]
48
    w2:
49
    w3:
50
51
    w1000:
52
53
   First transformation.
54
   flatten with value,
55
    apple -> ap
    apple -> pp
56
57
    apple -> pl
58
    apple -> le
    apple -> app
59
    apple -> ppl
60
61
    apple -> ple
62
63
64
```

```
65
 66
     Tranformation group by value,
 67
     ap -> [(apple -> ap), (appliation -> ap), ...]
 68
     pp -> [(), ()]
 69
 70
 71
 72
 73
 74
     map apply to values tuple
     ap -> [apple, application, . . .], distinct.
 75
 76
 77
     This is inverted index.
 78
     idf -> N/count values. [Normalization.]
 79
 80
     [1, 0, 5, 6, 9, 0, 0, 8]
     [0, 0, 0, 0, 100]
[0, 0, 0, 1, 1, 0, 0, 0]
 81
 82
 83
 84
 85
     n \rightarrow input \rightarrow f(n) \rightarrow ? to compute.
 86
 87
     f:hash(n) => K constant time.
 88
     formula, -> K, retrieve using k. O(1)
 89
     f:max(n) \rightarrow linear \rightarrow O(n)
 90
     sorting -> O(nlogn) quick sort, merge sort, n^2
 91
 92
     finding index in list(vector):
 93
     ["Minu", "Astha", "Shankar", "Ayush", "Saurab"]
 94
 95
     query("Astha") -> 1
 96
     query("Santa") -> -1
 97
     O(n)
 98
     _____
99
     binary search
100
     O(logn)
101
     _____
102
     HashSet -> ? unordered. isPresent? O(1)
103
     Binary search -> O(logn)
104
     Seq. search \rightarrow O(n)
105
106
107
     n log n
108
     _____
     Fib Series
109
110
     _____
111
112
     Give me nth Fib number ?
113
     -----
114
     N^2
115
     NlongN
116
     _____
117
     normal recursion. n^2
118
119
     Memomization
120
     _____
121
122
     ----- You can check it.
123
124
    Astha
125
     _____
126
127
     [a, b, c, a, b, a]
128
     trainsform to
```

```
[a \rightarrow 1, b \rightarrow 1, c \rightarrow 1, a \rightarrow 1, b \rightarrow 1, a \rightarrow 1]
129
130
131
     group_by
     first key,
132
133
134
      [a [1, 1, 1], b [1, 1], c [1], d [1]]
135
      sum value.
136
137
     [a 3, b 2, c 1, d 1]
138
139
     Total data N
140
      3 N
141
     O(N) \Rightarrow linear time.
142
143
      -----
144
     zip function
145
     -----
146
     x = [a, b, a, b, a, a]
     y = [1, 1, 1, 1, 1, 1]
147
148
149
     zip function
150
      zip(x,y) \Rightarrow [(a, 1), (b, 1), (a, 1), (b, 1) \dots]
151
152
153
     [a, b]
154
     [0 a, 1 b] \Rightarrow hashmap
155
      -----
156
    Binod
157
158
         1. trying to learn data transformation with other data.
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
```